BLAST Algorithm

See also:

- **BLAST Algorithm**
- **Smith-Waterman Algorithm**
- **Needleman-Wunsch Algorithm**

BLAST	
Developed by	Myers, E., Altschul S.F., Gish W., Miller E.W., Lipman D.J., NCBI
Latest release	2.2.18
OS	UNIX, Linux, Mac, MS-Windows
Genre	Bioinformatics tool
License	Public Domain

In bioinformatics, **Basic Local Alignment Search Tool**, or **BLAST**, is an algorithm for comparing primary biological sequence information, such as the amino-acid sequences of different proteins or the nucleotides of DNA sequences. A *BLAST search* enables a researcher to compare a query sequence with a library or database of sequences, and identify library sequences that resemble the query sequence above a certain threshold. For example, following the discovery of a previously unknown gene in the mouse, a scientist will typically perform a BLAST search of the human genome to see if humans carry a similar gene; BLAST will identify sequences in the human genome that resemble the mouse gene based on similarity of sequence. The BLAST program was designed by Eugene Myers, Stephen Altschul, Warren Gish, David J. Lipman and Webb Miller at the NIH and was published in J. Mol. Biol. in 1990[1].

Website	ftp://ftp.ncbi.nlm.nih.gov/blast/

Background

BLAST is one of the most widely used bioinformatics programs[2], because it addresses a fundamental problem and the algorithm emphasizes speed over sensitivity. This emphasis on speed is vital to making the algorithm practical on the huge genome databases currently available, although subsequent algorithms can be even faster.

Before the fast algorithm such as BLAST and FASTA were developed, doing database searches for the protein or nucleic sequences was very time consuming by using a full alignment program like dynamic programming. BLAST is about 50 times faster than the dynamic programming; however, it can not guarantee the optimal alignments of the query and database sequences as in the dynamic programming, but just work to find the related sequences in a database search. BLAST is more time efficient than FASTA by searching only for the more significant patterns in the sequences, but with comparative sensitivity. This could be further realized by knowing the algorithm of BLAST introduced below.

Examples of other questions that researchers use BLAST to answer are:

- Which bacterial species have a protein that is related in lineage to a certain protein with known amino-acid sequence?
- Where does a certain sequence of DNA originate?
- What other genes encode proteins that exhibit structures or motifs such as ones that have just been determined?

BLAST is also often used as part of other algorithms that require approximate sequence matching.

The BLAST algorithm and the computer program that implements it were developed by Stephen Altschul, Warren Gish, David Lipman at the U.S. National Center for Biotechnology Information (NCBI), Webb Miller at the Pennsylvania State University, and Gene Myers at the University of Arizona. It is available on the web on the NCBI website. Alternative implementations include WU-BLAST and FSA-BLAST.

The original paper by Altschul, *et al.*[1] was the most highly cited paper published in the 1990s.[3]

Input/Output

Input and output conform to the FASTA format.

Algorithm

BLAST

To run, BLAST requires a query sequence to search for, and a sequence to search against (or a sequence database containing multiple such sequences)(also called the target sequence). BLAST will find subsequences in the database which are similar to subsequences in the query. In typical usage, the query sequence is much smaller than the database, e.g., the query may be one thousand nucleotides while the database is several billion nucleotides.

The main idea of BLAST is that there are often high-scoring segment pairs (HSP) contained in a statistically significant alignment. BLAST searches for high scoring sequence alignments between the query sequence and sequences in the database using a heuristic approach that approximates the Smith-Waterman algorithm. The exhaustive Smith-Waterman approach is too slow for searching large genomic databases such as GenBank. Therefore, the BLAST algorithm uses a heuristic approach that is less accurate than the Smith-Waterman but over 50 times faster. The speed and relatively good accuracy of BLAST are among the key technical innovation of the BLAST programs.

Here the algorithm of BLASTP (a protein to protein search) is introduced to present the concept of BLAST.[4]

1. **Remove low-complexity region or sequence repeats in the query sequence.**
   Low-complexity region means a region of a sequence is composed of few

kinds of elements. These regions might give high scores that confuse the program to find the actual significant sequences in the database, so they should be filtered out. The regions will be marked with an X (protein sequences) or N (nucleic acid sequences) and then be ignored by the BLAST program. To filter out the low-complexity regions, the SEG program is used for protein sequences and the program DUST is used for DNA sequences. On the other hand, the program XNU is used to mask off the tandem repeats in protein sequences.

2. **Make a k-letter word list of the query sequence.**
   Take k=3 for example, we list the words of length 3 in the query protein sequence (k is usually 11 for a DNA sequence) "sequentially", until the last letter of the query sequence is included. The method can be illustrated in figure 1.
3. **List the possible matching words.**
   This step is one of the main differences between BLAST and FASTA. FASTA cares about all of the common words in the database and query sequences that are listed in step 2; however, BLAST cares about only the high-scoring words. The scores are created by comparing the word in the list in step 2 with all the 3-letter words. By using the scoring matrix (substitution matrix) to score the comparison of each residue pair, there are 20^3 possible match scores for a 3-letter word. For example, the score obtained by comparing PQG with PEG and PQA is 15 and 12, respectively. For DNA words, a match is scored as +5 and a mismatch as -4. After that, a neighborhood word score threshold T is used to reduce the number of possible matching words. The words whose scores are greater than the threshold T will remain in the possible matching words list, while those with lower scores will be discarded. For example, PEG is kept, but PQA is abandoned when T is 13.

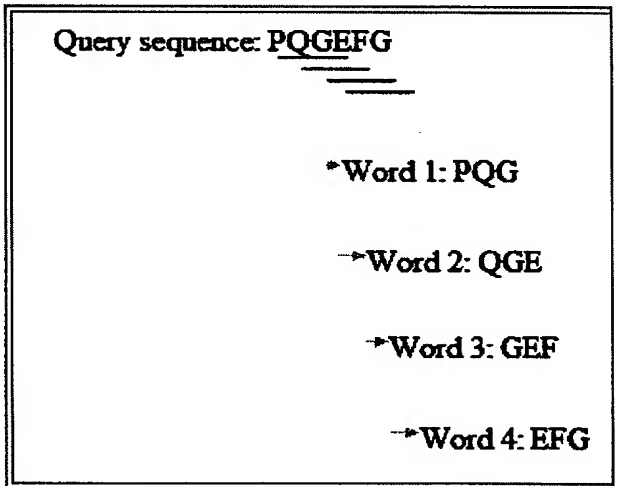


Fig. 1 The method to establish the k-letter query word list.

4. **Organize the remaining high-scoring words into an efficient search tree.**
   This is for the purpose that the program can rapidly compare the high-scoring words to the database sequences.
5. **Repeat step 1 to 4 for each 3-letter word in the query sequence.**
6. **Scan the database sequences for exact match with the remaining high-scoring words.**

The BLAST program scans the database sequences for the remaining high-scoring word, such as PEG, of each position. If an exact match is found, this match is used to seed a possible ungapped alignment between the query and database sequences.

7. **Extend the exact matches to high-scoring segment pair (HSP).**
   - The original version of BLAST stretches a longer alignment between the query and the database sequence in left and right direction, from the position where exact match is scanned. The extension doesn’t stop until the accumulated total score of the HSP begins to decrease. A simplified example is presented in figure 2.
   - To save more time, a newer version of BLAST, called BLAST2 or gapped BLAST, has been developed. BLAST2 adopts a lower neighborhood word score threshold to maintain the same level of sensitivity for detecting sequence similarity. Therefore, the possible matching words list in step 3 becomes longer. Next, the exact matched regions, within distance A from each other on the same diagonal in figure 3, will be joined as a longer new region. Finally, the new regions are then extended as the same method in the original version of BLAST, and the HSPs’ (High-scoring segment pair) scores of the extended regions are then created by using a substitution matrix as before.

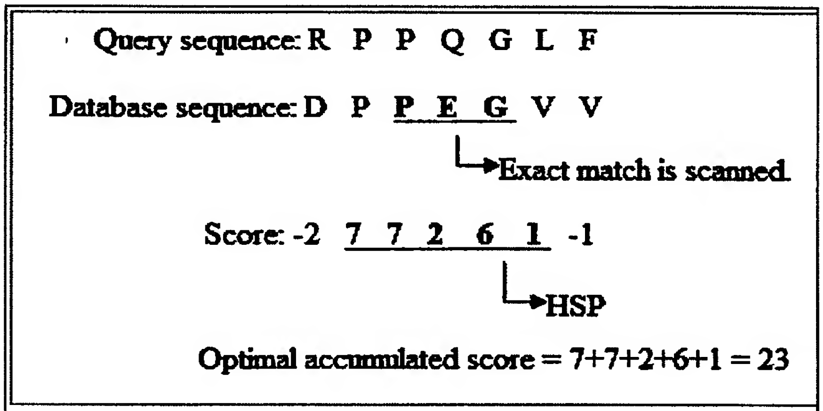
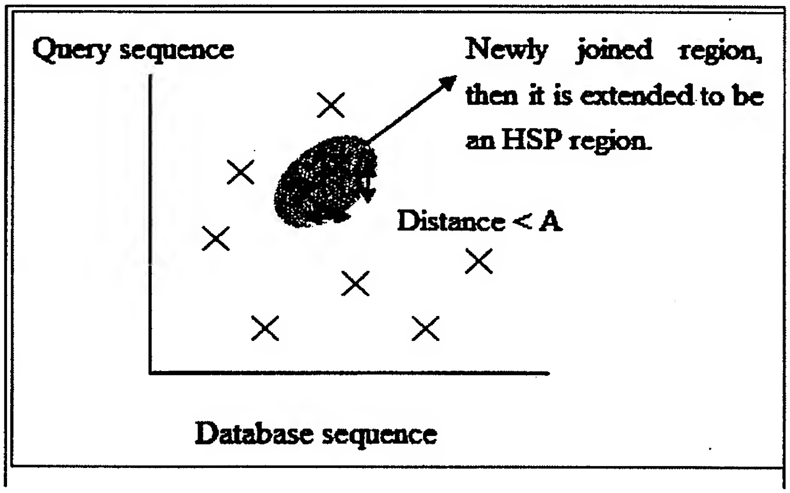

Query sequence: R P P Q G L F
Database sequence: D P P E G V V
Exact match is scanned.
Score: -2 7 7 2 6 1 -1
HSP
Optimal accumulated score = 7+7+2+6+1 = 23

Fig. 2 The process to extension the exact match.

8. **List all of the HSPs in the database whose score is high enough to be considered.**
   We list the HSPs whose scores are greater than the empirically determined cutoff score S. By examining the distribution of the alignment scores modeled by comparing random sequences, a cutoff


Query sequence
Newly joined region, then it is extended to be an HSP region.
Distance < A
Database sequence

score S can be determined such that its value is large enough to guarantee the significance of the remained HSPs.

Fig. 3 The positions of the exact matches.

9. **Evaluate the significance of the HSP score.**
BLAST next assesses the statistical significance of each HSP score by exploiting the Gumbel extreme value distribution (EVD). (It is proved that the distribution of Smith-Waterman local alignment scores between two random sequences follows the Gumbel EVD, regardless of whether gaps are allowed in the alignment). In accordance with the Gumbel EVD, the probability p of observing a score S equal to or greater than x is given by the equation
$$p(S \geq x) = 1 - \exp\left(-e^{-\lambda(x-\mu)}\right)$$
,where
$$\mu = {}^{[\log(Km'n')]}\!/_{\lambda}$$
The statistical parameters λ and K are estimated by fitting the distribution of the ungapped local alignment scores, of the query sequence and a lot of shuffled versions (Global or local shuffling) of a database sequence, to the Gumbel extreme value distribution. Note that λ and K depend upon the substitution matrix, gap penalties, and sequence composition (the letter frequencies).The m' and n' is the effective length of the query and database sequence, respectively. The original sequence length is shortened to the effective length to compensate for the edge effect (an alignment start near the end of one of the query or database sequence is likely not to have enough sequence to build an optimal alignment). They can be calculated as
$$m' \approx m - {}^{(\ln Kmn)}\!/_{H}$$
$$n' \approx n - {}^{(\ln Kmn)}\!/_{H},$$
where H is the average expected score per aligned pair of residues in an alignment of two random sequences. Altschul and Gish gave the typical values, λ = 0.318, K = 0.13, and H = 0.40, for ungapped local alignment using BLOSUM62 as the substitution matrix. Using the typical values for assessing the significance is called the lookup table methods, it is not accurate.The expect score E of a database match is the number of times that an unrelated database sequence would obtain a score S higher than x by chance. The expectation E obtained in a search for a database of D sequences is given by
$$E \approx 1 - e^{-p(s>x)D}$$
Furthermore, when $p < 0.1$, E could be approximated by the Poisson distribution as
$$E \approx pD$$
Note that the E value accessing the significance of the HSP score here (for ungapped local alignment) is not identical to the one in the later step to

evaluate the final gapped local alignment score, due to the variation of the statistical parameters.

10. **Make two or more HSP regions into a longer alignment.**
    Sometimes, we find two or more HSP regions in one database sequence that can be made into a longer alignment. This provides additional evidence of the relation between the query and database sequence. There are two methods, the Poisson method and the sum-of scores method, to compare the significance of the newly combined HSP regions. Suppose that here are two combined HSP regions with the sets of score (65, 40) and (52, 45), respectively. The Poisson method gives more significance to the set with the lower score of each set is higher (45>40). However, the sum-of-scores method prefers the first set, because 65+40 (105) is greater than 52+45(97). The original BLAST uses the Poisson method; gapped BLAST and the WU-BLAST use the sum-of scores method.
11. **Show the gapped Smith-Waterman local alignments of the query and each of the matched database sequences.**
    - The original BLAST only generates ungapped alignments including the initially found HSPs individually, even when there is more than one HSP found in one database sequence.
    - BLAST2 versions produce a single alignment with gaps that can include all of the initially found HSP regions. Note that the computation of the score and its corresponding E score is involved with the adequate gap penalties.
12. **Report the matches whose expect score is lower than a threshold parameter E.**

Parallel BLAST

Parallel BLAST versions are implemented using MPI and Pthreads, and have been ported to various platforms including Windows, Linux, Solaris, Mac OS X, and AIX. Popular approaches to parallelize BLAST include query distribution, hash table segmentation, computation parallelization, and database segmentation (partition).

Program

The BLAST program can either be downloaded and run as a command-line utility "blastall" or accessed for free over the web. The BLAST web server, hosted by the NCBI, allows anyone with a web browser to perform similarity searches against constantly updated databases of proteins and DNA that include most of the newly sequenced organisms.

BLAST is actually a family of programs (all included in the blastall executable). These include:

Nucleotide-nucleotide BLAST (blastn)

This program, given a DNA query, returns the most similar DNA sequences from the DNA database that the user specifies.

Protein-protein BLAST (blastp)

This program, given a protein query, returns the most similar protein sequences from the protein database that the user specifies.

Position-Specific Iterative BLAST (PSI-BLAST)

This program is used to find distant relatives of a protein. First, a list of all closely related proteins is created. These proteins are combined into a general "profile" sequence, which summarises significant features present in these sequences. A query against the protein database is then run using this profile, and a larger group of proteins is found. This larger group is used to construct another profile, and the process is repeated.

By including related proteins in the search, PSI-BLAST is much more sensitive in picking up distant evolutionary relationships than a standard protein-protein BLAST.

Nucleotide 6-frame translation-protein (blastx)

This program compares the six-frame conceptual translation products of a nucleotide query sequence (both strands) against a protein sequence database.

Nucleotide 6-frame translation-nucleotide 6-frame translation (tblastx)

This program is the slowest of the BLAST family. It translates the query nucleotide sequence in all six possible frames and compares it against the six-frame translations of a nucleotide sequence database. The purpose of tblastx is to find very distant relationships between nucleotide sequences.

Protein-nucleotide 6-frame translation (tblastn)

This program compares a protein query against the all six frame translations of a nucleotide sequence database.

Large numbers of query sequences (megablast)

When comparing large numbers of input sequences via the command-line BLAST, "megablast" is much faster than running BLAST multiple times. It concatenates many input sequences together to form a large sequence before searching the BLAST database, then post-analyze the search results to glean individual alignments and statistical values.

Alternative versions

An extremely fast but considerably less sensitive alternative to BLAST that compares nucleotide sequences to the genome is BLAT (Blast Like Alignment

Tool). A version designed for comparing multiple large genomes or chromosomes is BLASTZ.

Accelerated versions

- There are two main field-programmable gate array (FPGA) implementations of the BLAST algorithm. Progeniq is up to 100x faster than a software implementation running on the same processor. TimeLogic [1] offers a FPGA BLAST package called Tera-BLAST.
- The Mitrion-C Open Bio Project is an ongoing effort to port blast to run on Mitrion FPGAs. It is available on SourceForge.

References

1. ^ [a] [b] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990). "Basic local alignment search tool". *J Mol Biol* **215** (3): 403–410. doi:10.1006/jmbi.1990.9999. PMID 2231712.
2. ^ Casey, RM (2005). "BLAST Sequences Aid in Genomics and Proteomics". *Business Intelligence Network*.
3. ^ Sense from Sequences: Stephen F. Altschul on Bettering BLAST. ScienceWatch July/August 2000.
4. ^ D.W. Mount (2004). "Bioinformatics: Sequence and Genome Analysis.". Cold Spring Harbor Press.

External links

- NCBI-BLAST website
- NCBI-BLAST Tutorial
- WU-BLAST - The original gapping BLAST with statistics, developed and maintained by Warren Gish at Washington University in St. Louis
- EBI's BLAST Services - EBI's main blast services page.
- FSA-BLAST - A new, faster but still accurate version of NCBI BLAST based on recently published algorithmic improvements
- NBIC mpiBLAST - Netherlands Bioinformatics Centre, running mpiBLAST
- PatternHunter - An alternative software which provides similar functionality to BLAST while claiming increased speed and sensitivity
- Parallel BLAST - A dual scheduling BLAST tested on the Blue Gene/L
- BLAST HOWTO at the Wikiomics bioinformatics wiki
- A/G BLAST - Implementation for PowerPC G4/G5 processors and Mac OS X, from Apple Computer's Advanced Computation Group and Genentech.

- STRAP The protein workbench STRAP contains a comfortable BLAST front-end with a cache for BLAST results
- KoriBlast is a reliable graphical environment dedicated to sequence data mining. KoriBlast combines Blast searches with advanced data management capabilities and a state-of-the-art graphical user interface.
- Using the Basic Local Alignment Search Tool (BLAST)

Site Map About Us

Comments and inquiries could be addressed to:
webmaster@juliantrubin.com

Home

Last updated: July 2008